# Proiectarea unei Baze de Date a unui cinema

## Realizat de: Copilău Andrei, grupa 132

**Cuprins:**

1. **Descrierea modelului**
2. **Constrangeri**
3. **Entitati**
4. **Relatii**
5. **Atribute**
6. **Diagrama Entitate-Relatie**
7. **Diagrama Concenptuala**
8. **Schema relationala**
9. **Realizarea normalizarilor**
10. **Crearea secventelor**
11. **Crearea si inserarea datelor**
12. **Cereri SQL**
13. **Actualizarea si suprimarea datelor**
14. **Descrierea modelului**

Prin acest paragraf voi prezenta modelul real al bazei de date pe care o voi crea, utilitatile acesteia cat si regulile de functionare. Proiectul meu consta in implementarea si exploatarea unei baze de date relationare a unui cinematograf. Cu ajutorul acestui model de date, voi putea gestiona, atat clientii care doresc sa isi rezerve bilete pentru a viziona un film intr-o sala de cinema si la un program anume, cat si angajatii care se ocupa de preluarea acestor rezervari si departamentele unde lucreaza acestia. In total voi avea 8 tabele dintre care unul va fi asociativ si va face legatura intre tabelele clienti, bilete si filme. Fiecare tabel va avea o implementare importanta deoarece va ajuta la stocarea eficienta si organizata a datelor.

Un client isi poate cumpara mai multe bilete care pot contine mai multe locuri, pentru un anumit film care va fi difuzat intr-o anumita sala, la o data anume. Dupa ce clientul isi alege biletul, este realizata o rezervare care ulterior va fi preluata de un angajat care va lucra pe un departament anume.

1. **Constrangeri**

Pentru a asigura eficienta si integritatea datelor, voi aplica diferite constrangeri asupra tabelelor:

* Entitatea FILME: constrangeri de unicitate pentru titlurile fiecarui film pentru a nu exista filme cu acelasi titlu, de verificare a anului de lansare pentru a ma asigura ca este un an valid(2022<anul<2025) si constrangere de not null pentru titlul filmului
* Entitatea SALI: constrangeri de unicitate pentru id-ul fiecarei sali pentru a nu exista sali cu acelasi numar si de verificare pentru capacitatea salii pentru a ma asigura ca este mai mare decat zero
* Entitatea PROGRAM: cheie straina catre tabelele filme si sali pentru a ma asigura ca fiecare program se refera la un film si la o sala existenta si constrangere de unicitate pentru program\_id pentru a nu exista filme care sa se difuzeze la aceeasi data si in aceeasi sala
* Entitatea CLIENTI: constrangere de unicitate pentru numarul de telefon al fiecarei persoane pentru a evita inregistrarea mai multor clienti cu aceleasi date si constrangere de not null pentru numele clientilor
* Entitatea BILETE: cheie straina catre tabelele clienti si filme pentru a ma asigura ca fiecare bilet este achizitionat de un client pentru un film existent si verificarea ca nr\_locuri achizitionate sa fie mai mic decat 10 si pretul sa fie mai mic decat 1000
* Entitatea REZERVARI: chei straine catre tabelele clienti, bilete, filme si sali pentru a ma asigura ca fiecare rezervare este realizata de un client existent in functie de un bilet anume si pentru un film din cadrul cinemaului care se difuzeaza intr-o sala anume si constrangere de not null pentru data de achizitionare pentru rezervarea facuta
* Entitatea ANGAJATI: existenta unei chei straine catre tabelul departamente pentru a evidentia faptul ca fiecare angajat lucreaza intr-un departament anume
* Entitatea DEPARTAMENTE: constrangere de unicitate pentru numele departamentului, evitand existenta de departamente cu acelasi nume

1. **Entitati**

Entitatile acestei baze de date sunt urmatoarele:

* FILME(entitate independenta) – toate filmele difuzate la cinema. Cheia primara este atributul film\_id
* SALI(entitate independenta) – toate locatiile din cadrul cinemaului unde sunt difuzate filme. Cheia primara este atributul sala\_id
* PROGRAM(entitate independenta) – programul fiecarui film pentru sala in care acesta se difuzeaza. Cheia primara este atributul program\_id
* CLIENTI(entitate independenta) – clientii care achizitioneaza bilete la filme. Cheia primara este atributul client\_id
* BILETE(entitate independenta) – fiecare bilet achizitionat de un client. Cheia primara este atributul bilet\_id
* REZERVARI(entitate ce depinde de 3 tabele) – fiecare rezervare facuta de un client si pentru un film anume. Cheia primara este compusa, formata din atributele client\_id, bilet\_id si film\_id
* ANGAJATI(entitate independenta) – fiecare angajat care lucreaza in cadrul acestui cinema. Cheia primara este atributul angajat\_id
* DEPARTAMENTE(entitate independenta) – departamentele in care lucreaza angajatii. Cheia primara este atributul department\_id

1. **Relatii**

Relatiile pe baza carora s-au constituit tabelele sunt urmatoarele:

* „se difuzeaza” – relatie care leaga entitatile FILME si SALI. Relatia are cardinalitatea minima egala cu 1(1) si cea maxima egala cu m(1).
* „au” – relatie care leaga entitatile SALI si PROGRAM. Relatia are cardinalitatea minima egala cu 1(1) si cea maxima egala cu m(1).
* „se ocupa” – relatie care leaga entitatile BILETE si ANGAJATI. Relatia are cardinalitatea minima egala cu 1(1) si cea maxima egala cu m(1).
* „lucreaza” – relatie care leaga entitatile ANGAJATI si DEPARTAMENTE. Relatia are cardinalitatea minima egala cu 1(1) si cea maxima egala cu m(1).
* “rezerva” – relatie de tip 3 care leaga entitatile CLIENTI, BILETE si FILME.

1. **Atribute**

Pentru fiecare entitate sunt prezentate atributele sale, incluzand tipul de date si eventuale constrangeri, valori implicite sau valori posibile.

* Entitatea FILME are urmatoarele atribute:

film\_id = variabila de tip intreg de lungime maxima 3, care reprezinta un cod unic pentru filmele din cadrul cinemaului

titlul = variabila de tip caractere de lungime maxima 50, care reprezinta un nume unic de identificare al fiecarui film si care este not null

gen = variabila de tip caractere de lungime maxima 20, care reprezinta genul filmului ales

an\_lansare = variabila de tip data calendaristica, care reprezinta anul in care s-a lansat filmul ales si verifica daca este un an valid(intre 2023 si 2024)

regizor = variabila de tip caractere de lungime maxima 50, care reprezinta numele regizorului

* Entitatea SALI are urmatoarele atribute:

sala\_id = variabila de tip intreg de lungime maxima 2, care reprezinta un identificator unic pentru fiecare sala din cinema

capacitate = variabila de tip intreg de lungime maxima 2, care reprezinta capacitatea fiecarei sali din cinema si verifica daca aceasta este mai mare decat 0

* Entitatea PROGRAM are urmatoarele atribute:

program\_id = variabila de tip caractere de lungime maxima 20, care reprezinta un nume unic specific pentru fiecare program al unui film dintr-o sala

sala\_id = variabila de tip intreg de lungime maxima 2, care reprezinta un identificator unic pentru fiecare sala din cinema

film\_id = variabila de tip intreg de lungime maxima 3, care reprezinta un cod unic pentru filmele din cadrul cinemaului

data\_incepere = variabila de tip data calendaristica, care reprezinta data la care incepe filmul din sala respectiva

* Entitatea CLIENTI are urmatoarele atribute:

client\_id = variabila de tip intreg de lungime maxima 3, care reprezinta un cod unic specific pentru fiecare client al cinemaului

nume = variabila de tip caractere de lungime maxima 50, care reprezinta numele fiecarui client si are conditia de not null

varsta = variabila de tip intreg de lungime maxima 3, care reprezinta varsta fiecarui client

nr\_telefon = variabila de tip caractere de lungime 10, care reprezinta un identificator unic pentru numarul de telefon al fiecarui client

* Entitatea BILETE are urmatoarele atribute:

bilet\_id = variabila de tip intreg de lungime maxima 4 care reprezinta un cod unic specific pentru biletul achizitionat de fiecare client

client\_id = variabila de tip intreg de lungime maxima 3, care reprezinta un cod specific pentru fiecare client al cinemaului

film\_id = variabila de tip intreg de lungime maxima 3, care reprezinta un cod unic pentru filmele din cadrul cinemaului

nr\_locuri = variabila de tip intreg de lungime maxima 2, care reprezinta numarul de locuri selectate de un client pentru a viziona un film si se verifica ca acest numar sa fie mai mic decat 10

pret = variabila de tip intreg de lungime maxima 3, care reprezinta pretul de achizitionare al unui bilet la un film si care trebuie verificat pentru a fi mai mic decat 1000

* Entitatea REZERVARI are urmatoarele atribute:

client\_id = variabila de tip intreg de lungime maxima 3, care reprezinta un cod specific pentru fiecare client al cinemaului

bilet\_id = variabila de tip intreg de lungime maxima 4, care reprezinta un cod unic specific pentru biletul achizitionat de fiecare client

film\_id = variabila de tip intreg de lungime maxima 3, care reprezinta un cod unic pentru filmele din cadrul cinemaului

sala\_id = variabila de tip intreg de lungime maxima 2, care reprezinta un identificator unic pentru fiecare sala din cinema

data\_achizitionare = variabila de tip data calendaristica, care reprezinta data cand a fost facuta rezervarea si achizitionat biletul la film, aceasta avand conditita de not null

* Entitatea ANGAJATI are urmatoarele atribute:

angajat\_id = variabila de tip intreg de lungime maxima 3, care reprezinta un identificator unic al fiecarui angajat din cadrul cinemaului

department\_id = variabila de tip intreg de lungime maxima 2, care reprezinta un cod al fiecarui departament

nume\_angajat = variabila de tip caractere de lungime maxima 50, care reprezinta numele fiecarui angajat

specializare = variabila de tip caractere de lungime maxima 20, care reprezinta numele specializarii fiecarui angajat

salariu = variabila de tip intreg de lungime maxima 5, care reprezinta valoarea salariului obtinut de un angajat

hire\_date = variabile de tip data calendaristica, care reprezinta data de angajare a fiecarui angajat

* Entitatea DEPARTAMENTE are urmatoarele atribute:

department\_id = variabila de tip intreg de lungime maxima 2, care reprezinta un cod unic al fiecarui departament

nume\_departament = variabila de tip caractere de lungime maxima 20, care reprezinta un identificator unic al numelui fiecarui departament

1. **Diagrama entitate-relatie:**

FILME

film\_id#

titlul

gen

an\_lansare

regizor

SALI

sala\_id#

capacitate

PROGRAM

program\_id#

sala\_id

film\_id

data\_incepere

CLIENTI

client\_id#

nume

varsta

nr\_telefon

BILETE

bilet\_id#

client\_id

film\_id

nr\_locuri

pret

ANGAJATI

angajat\_id#

department\_id

nume\_angajat

specializare

salariu

hire\_date

DEPARTAMENTE

department\_id#

nume\_departament

se difuzeaza

au

se ocupa

lucreaza

m(1)

1

1

m(1)

m(1)

1

m(1)

1

m(1)

rezerva

m(1)

m(1)

1. **Diagrama conceptuala:**

FILME

film\_id#

titlul

gen

an\_lansare

regizor

SALI

sala\_id#

capacitate

PROGRAM

program\_id#

sala\_id

film\_id

data\_incepere

CLIENTI

client\_id#

nume

varsta

nr\_telefon

BILETE

bilet\_id#

client\_id

film\_id

nr\_locuri

pret

ANGAJATI

angajat\_id#

department\_id

nume\_angajat

specializare

salariu

hire\_date

DEPARTAMENTE

department\_id#

nume\_departament

REZERVARI

bilet\_id#

client\_id#

film\_id#

sala\_id

data\_achizitionare

1. **Schema relationala**

In continuare vor fi enumerate schemele relationale corespunzatoare diagramei conceptuale de la punctul 7, avand pentru fiecare entitate atributele sale, unele fiind urmate de semnul ‚#’ pentru a se evidentia cheia primara:

• FILME(film\_id#, titlul, gen, an\_lansare, regizor)

• SALI(sala\_id#, capacitate)

• PROGRAM(program\_id#, sala\_id, film\_id, data\_incepere)

• CLIENTI(client\_id#, nume, varsta, nr\_telefon)

• BILETE(bilet\_id#, client\_id, film\_id, nr\_locuri, pret)

• REZERVARI(bilet\_id#, client\_id#, film\_id#, sala\_id, data\_achizitionare)

• ANGAJATI(angajat\_id#, department\_id, nume\_angajat, specializare, salariu, hire\_date)

• DEPARTAMENTE(department\_id#, nume\_departament)

1. **Realizarea normalizarilor:**

--Forma FN1

Diagrama realizata se afla in forma normala FN1 deoarece fiecarui atribut al tabelelor ii corespunde o valoare indivizibila. Un exemplu de forma NON-FN1 ar putea sa existe in tabelul CLIENTI:

|  |  |
| --- | --- |
| #client\_id | date |
| 1 | Andrei, 0771469060 |
| 2 | Amalia, 0754321567 |

Se observa ca in exemplul anterior, coloana date se poate desparti in doua coloane distincte. Modelul corect de FN1 este urmatorul:

|  |  |  |
| --- | --- | --- |
| #client\_id | nume | telefon |
| 1 | Andrei | 0771469060 |
| 2 | Amalia | 0754321567 |

--Forma FN2

Modelul se afla in forma normala FN2 deoarece se afla in forma FN1 si fiecare atribut care nu este cheie primara este dependent de intreaga cheie primara. Un exemplu de forma NON-FN2 poate exista in tabelul REZERVARI:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| bilet\_id# | client\_id# | film\_id# | sala\_id | capacitate\_sala |
| 1 | 1 | 1 | 1 | 50 |
| 2 | 2 | 3 | 2 | 60 |

Se poate observa dependenta dintre cheia primara bilet\_id, client\_id, film\_id si atributul sala\_id, pe cand atributul capacitate\_sala este dependent doar de sala\_id. Prin urmare, modelul nu este in forma FN2. Corectand acest fapt, modelul corect de forma FN2 este urmatorul:

|  |  |  |  |
| --- | --- | --- | --- |
| bilet\_id# | client\_id# | film\_id# | sala\_id |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 3 | 2 |

--Forma FN3

Modelul creat se afla in forma normala FN3 deoarece este deja in forma FN2 si fiecare atribut care nu este cheie nu depinde de alt atribut, ci doar de cheia insasi. Un exemplu de forma NON-FN3 ar putea sa apara in tabelul FILME:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| film\_id# | titlul | gen | an\_lansare | regizor | nume\_sala |
| 2 | Forrest Gump | Drama | 15-MAY-23 | Robert Zeemekis | Distractie |
| 3 | Inception | SF | 20-JUL-23 | Christopher Nolan | Aventura |

Observam ca atributul nume\_sala nu este dependent de cheia primara film\_id, prin urmare tabelul nu este in FN3. Modelul corect de FN3 este urmatorul:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| film\_id# | titlul | gen | an\_lansare | regizor |
| 2 | Forrest Gump | Drama | 15-MAY-23 | Robert Zeemekis |
| 3 | Inception | SF | 20-JUL-23 | Christopher Nolan |

1. **Crearea secventelor pentru inserarea inregistrarilor:**

--secventa pentru incrementarea atributului sala\_id din tabelul sali

CREATE SEQUENCE seq\_sala

INCREMENT by 1

START WITH 1

MAXVALUE 10

NOCYCLE;

--secventa pentru incrementarea atributului angajat\_id din tabelul angajati

CREATE SEQUENCE seq\_angajat

INCREMENT by 1

START WITH 1

MAXVALUE 100

NOCYCLE;

--secventa pentru incrementarea atributului department\_id din tabelul departamente

CREATE SEQUENCE seq\_departament

INCREMENT by 1

START WITH 1

MAXVALUE 10

NOCYCLE;

**O imagine care conține text, captură de ecran, Font, număr

Descriere generată automat**

1. **Crearea si inserarea datelor:**

* Tabela CLIENTI

CREATE TABLE clienti(

client\_id NUMBER(3) PRIMARY KEY,

nume VARCHAR(50) NOT NULL,

varsta NUMBER(3),

nr\_telefon VARCHAR(10) UNIQUE

);

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (1, 'Andrei', 20, '0771469060');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (2, 'Amalia', 22, '0758932147');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (3, 'Catalin', 45, '0726415983');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (4, 'Mariana', 47, '0743058692');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (5, 'Bogdan', 16, '0779263815');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (6, 'Alex', 7, '0715472098');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (7, 'Karina', 17, '0738106429');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (8, 'Nicoleta', 41, '0762148359');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (9, 'Ema', 35, '0719835443');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (10, 'Antonio', 26, '0723459809');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (11, 'Rares', 20, '0773572100');

INSERT INTO clienti(client\_id, nume, varsta, nr\_telefon) VALUES (12, 'Marius', 55, '0727301854');

O imagine care conține text, captură de ecran, număr, Font

Descriere generată automat

* Tabela BILETE

CREATE TABLE bilete(

bilet\_id NUMBER(4) PRIMARY KEY,

client\_id NUMBER(3),

film\_id NUMBER(3),

nr\_locuri NUMBER(2) CHECK (nr\_locuri<10),

pret NUMBER(3) CHECK (pret<1000),

FOREIGN KEY (client\_id) REFERENCES clienti(client\_id),

FOREIGN KEY (film\_id) REFERENCES filme(film\_id)

);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (1, 1, 1, 1, 20);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (2, 2, 3, 2, 40);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (3, 3, 4, 1, 30);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (4, 4, 5, 4, 120);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (5, 5, 7, 2, 40);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (6, 6, 10, 1, 20);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (7, 7, 9, 3, 60);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (8, 8, 2, 2, 60);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (9, 9, 6, 1, 30);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (10, 10, 8, 5, 150);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (11, 11, 5, 3, 60);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (12, 12, 10, 1, 30);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (13, 1, 9, 2, 40);

INSERT INTO bilete(bilet\_id, client\_id, film\_id, nr\_locuri, pret) VALUES (14, 3, 6, 1, 30);

O imagine care conține text, captură de ecran, număr, Font

Descriere generată automat

* Tabela FILME

CREATE TABLE filme(

film\_id NUMBER(3) PRIMARY KEY,

titlul VARCHAR(50) NOT NULL,

gen VARCHAR(20),

an\_lansare DATE,

regizor VARCHAR(50),

CHECK (an\_lansare>=to\_date('2023-01-01', 'YYYY-MM-DD') AND an\_lansare<=to\_date('2024-12-31', 'YYYY-MM-DD')));

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (1, 'Forrest Gump', 'Drama', to\_date('2023-05-15', 'YYYY-MM-DD'), 'Robert Zemekis');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (2, 'Inception', 'SF', to\_date('2023-07-20', 'YYYY-MM-DD'), 'Christopher Nolan');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (3, 'Interstellar', 'SF', to\_date('2023-03-28', 'YYYY-MM-DD'), 'Christopher Nolan');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (4, 'Shutter Island', 'Mister', to\_date('2023-10-30', 'YYYY-MM-DD'), 'Martin Scorsese');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (5, 'Fast Furious', 'Actiune', to\_date('2023-05-26', 'YYYY-MM-DD'), 'Louis Leterrier');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (6, 'Se7en', 'Thriller', to\_date('2023-08-12', 'YYYY-MM-DD'), 'David Fincher');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (7, 'The Godfather', 'Actiune', to\_date('2023-09-01', 'YYYY-MM-DD'), 'Francis Ford Coppola');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (8, 'Matrix', 'SF', to\_date('2023-11-22', 'YYYY-MM-DD'), 'Lilly Wackowski');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (9, 'Dune', 'SF', to\_date('2023-07-07', 'YYYY-MM-DD'), 'Denis Villeneuve');

INSERT INTO filme(film\_id, titlul, gen, an\_lansare, regizor) VALUES (10, 'Bad Boys', 'Comedie', to\_date('2023-12-16', 'YYYY-MM-DD'), 'Michael Bay');

O imagine care conține text, captură de ecran, software, număr

Descriere generată automat

* Tabela REZERVARI

CREATE TABLE rezervari(

bilet\_id NUMBER(4),

client\_id NUMBER(3),

film\_id NUMBER(3),

sala\_id NUMBER(2),

data\_achizitionare DATE NOT NULL,

FOREIGN KEY (bilet\_id) REFERENCES bilete(bilet\_id),

FOREIGN KEY (client\_id) REFERENCES clienti(client\_id),

FOREIGN KEY (film\_id) REFERENCES filme(film\_id),

FOREIGN KEY (sala\_id) REFERENCES sali(sala\_id),

PRIMARY KEY(client\_id, bilet\_id, film\_id) --cheie primara compusa

);

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (1, 1, 1, 1, to\_date('2023-05-19', 'YYYY-MM-DD'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (2, 2, 3, 2, to\_date('2023-03-28', 'YYYY-MM-DD'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (3, 3, 4, 2, to\_date('2023-11-01', 'YYYY-MM-DD'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (4, 4, 5, 3, to\_date('2023-05-27', 'YYYY-MM-DD'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (5, 5, 7, 4, to\_date('2023-05-16', 'YYYY-MM-DD'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (6, 6, 10, 5, to\_date('2023-12-18', 'YYYY-MM-DD'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (7, 7, 9, 5, to\_date('2023-07-07', 'YYYY-MM-DD'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (8, 8, 2, 1, to\_date('2023-06-25', 'YYYY-MM-DD'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (9, 9, 6, 3, to\_date('2023-08-14', 'YYYY-MM-DD'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (10, 10, 8, 4, to\_date('2023-11-23', 'YYYY-MM-DD'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (11, 11, 5, 3, to\_date('2023-05-26', 'YYYY-MM-DD'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (12, 12, 10, 5, to\_date('2023-12-18', 'YYYY-MM-DD'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (13, 1, 9, 5, to\_date('2023-07-27', 'YYYY-MM-DD'));

INSERT INTO rezervari(bilet\_id, client\_id, film\_id, sala\_id, data\_achizitionare) VALUES (14, 3, 6, 3, to\_date('2023-08-28', 'YYYY-MM-DD'));

O imagine care conține text, captură de ecran, număr, software

Descriere generată automat

* Tabela DEPARTAMENTE

CREATE TABLE departamente(

department\_id NUMBER(2) PRIMARY KEY,

nume\_departament VARCHAR(20) UNIQUE

);

INSERT INTO departamente(department\_id, nume\_departament) VALUES (seq\_departament.nextval, 'vanzare');

INSERT INTO departamente(department\_id, nume\_departament) VALUES (seq\_departament.nextval, 'management');

INSERT INTO departamente(department\_id, nume\_departament) VALUES (seq\_departament.nextval, 'resurse umane');

INSERT INTO departamente(department\_id, nume\_departament) VALUES (seq\_departament.nextval, 'marketing');

INSERT INTO departamente(department\_id, nume\_departament) VALUES (seq\_departament.nextval, 'ingrijire');

O imagine care conține text, captură de ecran, software, Font

Descriere generată automat

* Tabela ANGAJATI

CREATE TABLE angajati(

angajat\_id NUMBER(3) PRIMARY KEY,

department\_id NUMBER(2),

nume\_angajat VARCHAR(50),

specializare VARCHAR(20),

salariu NUMBER(5),

hire\_date DATE,

FOREIGN KEY (department\_id) REFERENCES departamente(department\_id)

);

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 2, 'Copilau Andrei', 'manager', 10000, to\_date('2020-02-14', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 2, 'Florentin Rus', 'asistent management', 9000, to\_date('2020-02-17', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 2, 'Alex Constantin', 'asistent management', 8500, to\_date('2022-02-20', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 1, 'Kobel Dan', 'casier', 3000, to\_date('2021-07-07', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 1, 'Florentin Rus', 'casier', 3000, to\_date('2021-07-07', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 1, 'Florentin Rus', 'casier', 3500, to\_date('2020-03-02', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 1, 'Florentin Rus', 'casier', 4000, to\_date('2020-02-21', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 3, 'Dumitru Beatrice', 'specialist HR', 5500, to\_date('2020-02-22', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 3, 'Done Albert', 'specialist HR', 5500, to\_date('2021-02-22', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 4, 'Gheorghe Mariana', 'brand manager', 6000, to\_date('2020-05-13', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 4, 'Nichifor Amalia', 'brand manager', 6500, to\_date('2020-02-17', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 5, 'Ioana Stan', 'ingijitor', 3000, to\_date('2020-02-25', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 5, 'Serghei Florin', 'ingijitor', 2500, to\_date('2020-03-01', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 5, 'Ciochina Adriana', 'ingijitor', 2500, to\_date('2021-02-14', 'YYYY-MM-DD'));

INSERT INTO angajati(angajat\_id, department\_id, nume\_angajat, specializare, salariu, hire\_date) VALUES (seq\_angajat.nextval, 5, 'Alexa Andra', 'ingijitor', 2500, to\_date('2021-02-14', 'YYYY-MM-DD'));

O imagine care conține text, captură de ecran, număr, software

Descriere generată automat

* Tabela SALI

CREATE TABLE sali(

sala\_id NUMBER(2) PRIMARY KEY,

capacitate NUMBER(2) CHECK (capacitate>0)

);

INSERT INTO sali(sala\_id, capacitate) VALUES (seq\_sala.nextval, 50);

INSERT INTO sali(sala\_id, capacitate) VALUES (seq\_sala.nextval, 50);

INSERT INTO sali(sala\_id, capacitate) VALUES (seq\_sala.nextval, 60);

INSERT INTO sali(sala\_id, capacitate) VALUES (seq\_sala.nextval, 65);

INSERT INTO sali(sala\_id, capacitate) VALUES (seq\_sala.nextval, 70);

O imagine care conține text, captură de ecran, număr, software

Descriere generată automat

* Tabela PROGRAM

CREATE TABLE program(

program\_id VARCHAR(20) PRIMARY KEY,

sala\_id NUMBER(2),

film\_id NUMBER(3),

data\_incepere DATE,

FOREIGN KEY (sala\_id) REFERENCES sali(sala\_id),

FOREIGN KEY (film\_id) REFERENCES filme(film\_id)

);

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#1', 1, 1, to\_date('2023-05-16', 'YYYY-MM-DD'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#2', 1, 2, to\_date('2023-06-20', 'YYYY-MM-DD'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#3', 2, 3, to\_date('2023-03-28', 'YYYY-MM-DD'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#4', 2, 4, to\_date('2023-10-30', 'YYYY-MM-DD'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#5', 3, 5, to\_date('2023-05-26', 'YYYY-MM-DD'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#6', 3, 6, to\_date('2023-08-12', 'YYYY-MM-DD'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#7', 4, 7, to\_date('2023-09-02', 'YYYY-MM-DD'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#8', 4, 8, to\_date('2023-11-22', 'YYYY-MM-DD'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#9', 5, 9, to\_date('2023-07-07', 'YYYY-MM-DD'));

INSERT INTO program(program\_id, sala\_id, film\_id, data\_incepere) VALUES ('#10', 5, 10, to\_date('2023-12-17', 'YYYY-MM-DD'));

O imagine care conține text, electronice, captură de ecran, afișaj

Descriere generată automat

1. **Cereri SQL**

Aceasta parte a proiectului include cererile sql create pentru a raspunde la cerintele de la exercitiul 12.

1. Subcereri sincronizate in care intervin cel putin 3 tabele

ENUNT: Afisarea nume client, numele filmului, numarul de locuri alese, pretul biletului si data de achizitionare a acestuia pentru clientii care au optat pentru un film care se incadreaza in categoria ‚Actiune’.

--In cadrul acestei cereri am folosit tabelele clienti, bilete, rezervari si filme pentru a obtine informatiile necesare aplicand subcereri sincronizate in cadrul expresiei where.

select c.nume as nume, f.titlul as titlul, b.nr\_locuri, b.pret, r.data\_achizitionare

from clienti c join rezervari r on c.client\_id=r.client\_id

join bilete b on r.bilet\_id=b.bilet\_id

join filme f on r.film\_id=f.film\_id

where c.client\_id in (select client\_id

from rezervari

where film\_id in (select film\_id

from filme

where gen='Actiune')

);

O imagine care conține text, captură de ecran, software, afișaj

Descriere generată automat

b,d,f) Subcereri nesincronizate în clauza FROM, ordonari si utilizarea functiilor NVL si DECODE (în cadrul aceleiasi cereri) si utilizarea a cel putin 1 bloc de cerere (clauza WITH).

ENUNT: Afisarea informatiilor despre angajati: nume, salariile lor(daca sunt null le modific si pun valoarea 3000) si numele departamentelor unde lucreaza fiecare(pe care le modific cu decode), ordonati descrescator dupa salariul fiecaruia.

--Am folosit un bloc de cerere pentru a organiza o parte din date intr-o noua tabela temporar creata angajati\_info, urmand ca aceste date sa le combin cu cele ale tabelei departamente pentru a obtine toate informatiile necesare, aplicand functiile nvl si decode pentru a schimba atat numele departamentelor, cat si salariul in caz ca acesta era null si organizand totuul descrescator dupa salariu.

with angajati\_info as(

select nume\_angajat, nvl(salariu, 3000) as salariu\_nou, department\_id

from angajati

)

select a.nume\_angajat, a.salariu\_nou, decode(d.nume\_departament, 'vanzare', 'sales', 'management', 'management', 'resurse umane', 'HR', 'marketing', 'branding', 'alt job') as numedept\_nou

from (select \* from angajati\_info) a join (select \* from departamente) d on a.department\_id=d.department\_id

order by salariu\_nou desc;

O imagine care conține text, captură de ecran

Descriere generată automat

e) Utilizarea a cel putin 2 functii pe siruri de caractere, 2 functii pe date calendaristice, a cel putin unei expresii CASE.

ENUNT: Afisarea tuturor clientilor care au optat pentru filme din categoria SF si concatenarea titlului fiecarui film cu secventa '-sf'.

--Am folosit in realizarea acestei cereri 2 functii pe siruri de caractere si o expresie case.

select r.client\_id, concat(f.titlul, '-sf') as titlu\_film, case when upper(f.gen)='SF' then 'Film science fiction' else 'Alt gen de film' end as gen\_film

from rezervari r join filme f on r.film\_id=f.film\_id

where upper(f.gen)='SF';

O imagine care conține text, software, Font, număr

Descriere generată automat

e) Utilizarea a cel putin 2 functii pe siruri de caractere, 2 functii pe date calendaristice, a cel putin unei expresii CASE.

ENUNT: Afisarea numelui fiecarui client care si a cumparat un bilet in a doua jumatate a anului 2023, bilet\_id-ul, data de achizitionare a biletului, data de lansare si perioada cand poate viziona filmul respectiv(max la o luna de la data lansarii).

--Am folosit in realizarea acestei cereri 2 functii pe date calendaristice.

select c.nume, r.bilet\_id, r.data\_achizitionare, f.an\_lansare as data\_lansare, add\_months(f.an\_lansare, 1) as perioada\_vizionare

from clienti c join rezervari r on c.client\_id=r.client\_id

join filme f on r.film\_id=f.film\_id

where r.data\_achizitionare between to\_date('2023-06-01', 'YYYY-MM-DD') and to\_date('2023-12-31', 'YYYY-MM-DD');

O imagine care conține text, număr, software, Font

Descriere generată automat

1. Grupari de date, functii grup, filtrare la nivel de grupuri cu subcereri nesincronizate(în clauza de HAVING) în care intervin cel putin 3 tabele (in cadrul aceleiasi cereri).

ENUNT: Afisare nume client, titlul filmului si pretul biletului doar pentru clientii care au pretul biletului mai mic decat media preturilor tuturor biletelor

--Am folosit in realizarea acestei cereri 3 tabele: clienti, bilete si filme pentru a selecta informatiile dorite, grupandu-le dupa aceste informatii si filtrandu-le cu ajutorul clauzei having pentru a avea doar biletele mai mici decat media tuturor biletelor.

select c.nume, f.titlul, b.pret

from clienti c join bilete b on c.client\_id=b.client\_id

join filme f on b.film\_id=f.film\_id

group by c.nume, f.titlul, b.pret

having b.pret<(select avg(pret) from bilete);

O imagine care conține text, captură de ecran, software, număr

Descriere generată automat

1. **Actualizarea si suprimarea datelor**

Urmatoarele cereri includ 3 operatii de actualizare si suprimare a datelor folosind subcereri.

1. Actualizarea salariului angajatilor din departamentul HR, folosind tabelele angajati si departamente

update angajati

set salariu = salariu\*1.10

where department\_id = (select department\_id

from departamente

where nume\_departament='resurse umane'

);

O imagine care conține text, captură de ecran, număr, software

Descriere generată automat

b) Reducerea biletelor cu 20% pentru clientii cu varsta mai mica de 22 de ani, folosind tabelele bilete si clienti

update bilete

set pret = pret\*0.80

where client\_id in (select client\_id

from clienti

where varsta<=22

);

O imagine care conține text, captură de ecran, număr, Paralel

Descriere generată automat

c) Stergerea biletelor din cadrul tabelei asociative rezervari pentru rezervarile facute pana in luna mai

delete from rezervari

where bilet\_id in (select bilet\_id

from rezervari

where data\_achizitionare <= to\_date('2023-05-31', 'YYYY-MM-DD')

);

O imagine care conține text, captură de ecran, software, număr

Descriere generată automat